CLAIMS

1. A method for allocating core processor bandwidth in a computer between routine processing and interrupt servicing, wherein the core processor includes an instruction queue mechanism staging instructions for processing, the method comprising:

detecting an interrupt service request;

inserting interrupt servicing instructions responsive to the interrupt service request into the instruction queue mechanism; and

processing the instructions within the instruction queue mechanism including the inserted interrupt servicing instructions.

2. The method of claim 1 in which the instruction queue mechanism includes an instruction cache and an instruction fetch unit for fetching instructions from the instruction cache, said processing being performed in such manner as to decode the instructions into micro-opcodes and execute the micro-opcodes in one or more out-of-order execution units.

3. The method of claim 2 which further comprises retiring the executed micro-opcodes including those micro-opcodes representing the inserted interrupt servicing instructions to the instruction cache.

4. The method of claim 3 wherein the executed micro-opcodes are retired to the instruction cache in order.

5. The method of claim 1 wherein said detecting is of plural interrupts, which further comprises prioritizing such plural interrupts and inserting one or more instances of such interrupt servicing instructions into the instruction queue mechanism in accordance with one or more predefined interrupt servicing allocation criteria.

6. The method of claim 5 wherein the one or more allocation criteria include the priority of the interrupts and the capacity of the processor to allocate bandwidth to interrupt servicing.

7. The method of claim 6 wherein said prioritizing is dynamically responsive to changing allocation criteria.

8. The method of claim 1 wherein core processor bandwidth is allocated to interrupt servicing without flushing the instruction queue mechanism.

9. The method of claim 1 in which said detecting is performed by an interrupt processor, which after said detecting further comprises:

at the interrupt processor determining whether the detected interrupt service request is of a priority meeting one or more defined high-priority criteria and if so then signaling the core processor to perform said inserting; and

at the core processor responding to said signaling by performing said inserting and said processing.

10. The method of claim 1 in which said detecting is performed by an interrupt processor, which at the core processor further comprises:

determining that a natural core processor context switch is imminent;

signaling the interrupt processor to make ready the highest priority interrupt service request; and

signaling the instruction queue mechanism to fetch the readied interrupt service request in advance of the context switch.

11. A digital processor for use in a computer supporting one or more hardware interrupt inputs, the processor comprising:

an instruction cache;

a fetch-and-decode unit, said fetch-and-decode unit fetching instructions from said instruction cache and decoding the instructions into micro-opcodes;

a dispatch-and-execute unit having one or more execution ports, said dispatch-and-execute unit scheduling and executing the micro-opcodes in said one or more execution ports and thereafter retiring the micro-opcodes back into the instruction cache; and

an interrupt-handling mechanism responsive to one or more hardware interrupt inputs, said interrupt-handling mechanism instructing said fetch-and-decode unit to insert into a normal instruction sequence decoded micro-opcodes representing interrupt servicing instructions for scheduling and execution by said dispatch-and-execute unit.

12. The apparatus of claim 11 in which the computer supports plural hardware interrupt inputs, wherein said interrupt-handling mechanism includes an interrupt concentrator for determining priority among the plural hardware interrupt inputs and for scheduling the plural hardware interrupt inputs, the interrupt-handling mechanism instructing said fetch-and-decode unit to insert plural instances of such decoded micro-opcodes, each instance representing one or more corresponding sets of interrupt servicing instructions, into the normal instruction sequence for serial scheduling and execution of the plural instances of such decoded micro-opcodes by said dispatch-and-execute unit in accordance with the determined priority of said plural hardware interrupt inputs.

13. The apparatus of claim 10 which further comprises:

a context switch prediction unit coupled with said fetch-and-decode unit predicting a naturally occurring context switch and for signaling said interrupt-handling mechanism upon such prediction,

said interrupt-handling mechanism responsive to such signaling from said prediction unit instructing said fetch-and-decode unit.

14. Apparatus for allocating processor bandwidth to hardware interrupts in a computer comprising:

an instruction queue mechanism staging p-code for execution;

plural interrupt signals for coupling to one or more hardware input/output devices;

interrupt priority logic for ranking the relative priority of plural interrupts represented by the plural interrupt signals;

bandwidth allocation logic for determining the timing by which a ranked one or more of the plural interrupts is to be serviced;

a p-code insertion mechanism for inserting interrupt servicing p-code in accordance with the determined timing into said instruction queue mechanism for processing; and

a p-code processor coupled with said instruction queue mechanism and responsive to said insertion mechanism, said p-code processor executing p-code within said instruction queue mechanism, the executed p-code including the interrupt servicing p-code.

15. The apparatus of claim 14, wherein said instruction queue mechanism includes an instruction cache and an in-order instruction fetch-and-decode unit performing branch predictions and wherein said p-code processor includes a dispatch and out-of-order execution unit scheduling and executing p-code in parallel among one or more execution ports.

16. The apparatus of claim 14 which further comprises an in-order retirement unit retiring instructions back to said instruction cache.

17. The apparatus of claim 15 wherein said fetch-and-decode unit includes said allocation logic.

18. The apparatus of claim 17 wherein said allocation logic is responsive to a computer operating system.

19. The apparatus of claim 14 wherein said priority logic is responsive to a computer operating system.

20. The apparatus of claim 14 wherein said allocation logic is responsive to a current-usage model.

21. The apparatus of claim 14 wherein said priority logic is responsive to a current-usage model.

22. An article of manufacture for use with a core processor including an instruction queue, the article comprising a computer-readable medium containing a program, the program comprising:

detection firmware for detecting an interrupt service request;

insertion firmware for inserting interrupt servicing instructions responsive to the interrupt service request into the instruction queue; and

processor firmware for processing the instructions within the instruction queue including the inserted interrupt servicing instructions.

23. A computer-readable medium containing a program according to claim 22 wherein the program further comprises:

signaling firmware coupled with said detection firmware for signaling said insertion firmware with a vector representing an interrupt service request upon a determination by said detection firmware that an interrupt service request of a priority that exceeds a given threshold has occurred,

said insertion firmware fetching interrupt service instructions from memory in accordance with the vector representing the determined priority interrupt service request.

24. A computer-readable medium containing a program according to claim 22 wherein the program further comprises:

signaling firmware coupled with said insertion firmware for signaling said detection firmware upon detection by said signaling firmware of an impending natural context switch,

detection of the impending natural context switch being predicated on an instruction stream within the instruction queue of the core processor.

25. A computer system comprising:

one or more input/output (I/O) devices each capable of generating one or more hardware interrupts;

a core processor coupled with said interrupt concentrator, said core processor including an instruction cache, a fetch-and-decode unit, said fetch-and-decode unit fetching instructions from said instruction cache and decoding the instructions into

micro-opcodes, said core processor further including a dispatch-and-execute unit having one or more execution ports, said dispatch-and-execute unit scheduling and executing the micro-opcodes in said one or more execution ports and thereafter retiring the micro-opcodes back into the instruction cache; and

an interrupt-handling mechanism responsive to said one or more hardware interrupts, said interrupt-handling mechanism instructing said fetch-and-decode unit to insert into a normal instruction sequence decoded micro-opcodes representing interrupt servicing instructions for scheduling and execution by said dispatch-and-execute unit.

26. The computer system of claim 25 which supports plural hardware interrupt inputs, wherein said interrupt-handling mechanism includes an interrupt concentrator for determining priority among and scheduling the plural hardware interrupts, the interrupt-handling mechanism instructing said fetch-and-decode unit to insert plural instances of such decoded micro-opcodes, each instance representing one or more corresponding sets of interrupt servicing instructions, into the normal instruction sequence for serial scheduling and execution of the plural instances of such decoded micro-opcodes by said dispatch-and-execute unit in accordance with the determined priority of said plural hardware interrupts.

27. The computer system of claim 25 which further comprises:

a context switch prediction unit coupled with said fetch-and-decode unit predicting a naturally occurring context switch and for signaling said interrupt-handling mechanism upon such prediction,

said interrupt-handling mechanism responsive to such signaling from said prediction unit instructing said fetch-and-decode unit.